IN THE SPECIFICATION

Please replace the paragraph beginning on page 2, line 10 and ending on page 2, line 13 with the following amended paragraph.

F1

A positive floating point ~~number~~variable x can be represented by an expression written as:

$$x = m \times 2^e \qquad (1)$$

where m ($1 \leq m < 2$) is a mantissa and e is a binary exponent.

Please replace the paragraph beginning on page 3, line 8 and ending on page 3, line 17 with the following amended paragraph.

F2

There is therefore provided, in one embodiment of the present invention, a method for computing a natural logarithm function that includes steps of: partitioning a mantissa region between 1 and 2 into N equally spaced sub-regions; precomputing centerpoints a_i of each of the N equally spaced sub-regions, where $i = 0,\ldots,N-1$; selecting N sufficiently large so that, for each sub-region, a first degree polynomial in m computes log(m) to within a preselected degree of accuracy for any m within the sub-region, where m is a mantissa of a binary floating point representation of a ~~number;~~variable; and computing a value of log(x) for a binary floating point representation of ~~a particular number~~ x stored in a memory of a computing device utilizing the first degree polynomial in m.

Please add the following paragraph after page 3 and before page 4, line 1.

F3

Figure 3 is a flowchart of an embodiment of a method for fast natural log(x) calculation.

Please replace the paragraph beginning on page 4, line 1 and ending on page 4, line 3 with the following amended paragraph.

F4

~~Figure 3~~Figure 4 is a representation of a ~~number~~variable stored in IEEE single-precision binary floating point format, partitioned as in one embodiment of the invention.

Please add the following paragraph after page 5, line 4 and before page 5, line 5.

F5

Figure 3 is a flowchart of an embodiment of a method for fast natural log(x) calculation. When executing the method, which is described in detail below, computer 36 partitions 62 a mantissa region between 1 and 2 into N equally spaced sub-regions and precomputes 64 a reference point a_i of each of the N equally spaced sub-regions, where $i = 0,\ldots,N-1$. Computer 36 selects 62 N sufficiently large so that, for each sub-region, a first degree polynomial in m computes $\log(m)$ to within a preselected degree of accuracy for any m within the sub-region, where m is a binary mantissa of a binary floating point representation of a variable x. Computer 36 calculates 66 a value of $\log(x)$ for a binary floating point representation of x stored in mass storage device 38 of computer 36 utilizing the first degree polynomial in m, where $\log(x)$ is a function of a distance between a_i and the mantissa. Image reconstructor 34 generates 68 an image by using the computed value of $\log(x)$.

Please replace the paragraph beginning on page 5, line 14 and ending on page 6, line 4 with the following amended paragraph.

F6

Because $(m\text{-}a)<1$, there are two ways to minimize the error. One way is to increase the order of the approximation, and the other is to minimize the distance from m to a. Because mantissa m is between 1 and 2, in one embodiment of the present invention, the

region between 1 and 2 is partitioned into N equally spaced sub-regions. Centers of each of the sub-regions are precomputed and used as reference points in equations (4a) and (4b). By partitioning into a sufficiently large number of sub-regions, a low order polynomial function produces sufficient accuracy for CT imaging purposes. In particular, by selecting a sufficiently large number of sub-regions, for any m within any particular sub-region, $\log(m)$ is computed by a first-degree polynomial to within a preselected degree of accuracy within that sub-region. For example, computer 36 uses the first degree polynomial in m to compute values of $\log(x)$ for binary floating point representations of ~~particular numbers~~ x stored in its memory.

F6

Please replace the paragraph beginning on page 6, line 9 and ending on page 6, line 26 with the following amended paragraph.

Rather than compute a sub-region index using $i = round((m-1) \times N)$, which would require six operations, one embodiment of the present invention reduces computation load as follows. A partitioning algorithm divides the mantissa of a binary floating point ~~number~~variable in memory into two sub-regions. The sub-regions have index i and Δx, where Δx is a distance from mantissa m to reference point a_i. Indices i and Δx are directly extracted from an IEEE floating-point ~~number~~variable stored in a computer system, thereby reducing computation time and improving accuracy. In one embodiment, mantissa partitioning occurs as illustrated in ~~Figure 3,~~Figure 4, in which index i ranges from 0 to 127 and each region represents information extracted from the datum shown in ~~Figure 3.~~Figure 4. More particularly, in a single precision IEEE floating point ~~number,~~variable, b_{31} represents a sign bit, b_{30} the most significant bit of exponent e, b_{23} the least significant bit of exponent e, b_{22} the most significant bit of mantissa m, and b_0 the least significant bit of mantissa m. (If it is desired to use a different designation for the numbering of bits b, those skilled in the art can make the appropriate changes required in the description for notational consistency.) In this single precision embodiment, exponent e is extracted directly from bits b_{30} to b_{23}; region i

F7

F7

is extracted directly from bits b_{22} to b_{16}; and Δx (a distance from mantissa m to reference point a_i) is extracted directly from bits b_{15} to b_0.

Please replace the paragraph beginning on page 7, line 1 and ending on page 7, line 3 with the following amended paragraph.

F8

Using the extraction illustrated in ~~Figure 3,~~Figure 4, a maximum error of equation (6) in each sub-region is estimated by an expression written as:

$$error \leq \frac{1}{2a_i^2} \times \left(\frac{1}{2N}\right)^2; \quad i = 0, \ldots, N-1; \quad 1 \leq a_i < 2 \qquad (7a)$$